7N-63-TM
91478
P-13

Controlling a Dynamic Physical System with Approximate Reasoning

HAMID BERENJI, YUNG-YAW CHEN, AND CHUEN-CHEIN LEE MURUGESAN

AI RESEARCH BRANCH, MAIL STOP 244-17
NASA AMES RESEARCH CENTER
MOFFETT FIELD, CA 94035

NASA Ames Research Center
Artificial Intelligence Research Branch

Technical Report RIA-88-12-05-1

December, 1988

Controlling a Dynamic Physical System with Approximate Reasoning*

Hamid Berenji†, Yung-Yaw Chen‡, Chuen-Chein Lee‡ S. Murugesan

NASA Ames Research Center
Mail Stop: 244-17, Moffett Field, CA 94035

Abstract

This paper presents an approach for controlling a dynamic physical system by using approximate reasoning. The approach has been implemented in a program named POLE and we have successfully built a prototype hardware system to solve the *cart-pole balancing* problem in real-time. This provides a complementary alternative to the conventional analytical control methodology, and is of substantial use where a precise mathematical model of the process being controlled is not available. Also, we furnish a set of criteria for comparing controllers based on approximate reasoning and those based on conventional control schemes.

Topic Areas: Design, Manufacturing, Control (C4), Real-Time Performance (A3).

*Supported partially by NASA Ames Director's Discretionary Fund

†Also with Heer Associates Inc.

‡Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, 94720

1 Introduction

Human expert controllers often perform superbly well under conditions of uncertainty and imprecision using mainly approximate reasoning. They select *control actions* based on a quick assesment of the process which they are controlling. In fact, learning to control a physical system has been regarded as one kind of *intelligence* [6]. Control theorists have successfully dealt with a large class of control problems by mathematically modeling the process and solving these analytical models to generate control actions. However, the analytical models tend to become complex, especially in large, intricate systems. The non-linear behavior of many practical systems makes this analytical approach even more difficult. In this paper, we illustrate how techniques from *Approximate Reasoning* and *Knowledge-Based Control* can be used to provide a viable alternative to the traditional analytical control.

In particular, we explore how qualitative control parameters can be handled by a technique known as *Fuzzy Control*, originally proposed by Mamdani and Assilian [8], who based their work on Zadeh's fuzzy set theory [15]. We compare fuzzy control with the state feedback control, a popular approach in modern digital control. Based on this comparison, we propose a set of criteria which can be used for evaluating and selecting a suitable technique for controlling non-linear and complex dynamic systems. This comparative study is made using computer simulations of a cart-pole balancing problem which represents a typical non-linear system. This interesting problem has served as a basis for study by many connectionists and control theorists (e.g., Shaefer and Cannon [13]) and can be considered as the *blocksworld* of control theory. Learning of the control process for this problem has been studied by Michie and Chambers [9], and by Selfridge, Sutton, and Barto [12]. In this learning research, the objective has been to write a program which can learn to keep the pole balanced.

The organization of this paper is as follows. With a brief overview of fuzzy control theory and rule-based control, we contrast rule-based controllers with the controllers based on conventional control theory, specifying their advantages and disadvantages. The development of the cart-pole balancing program (POLE) and its associated prototype hardware system is described next. Finally, we develop a set of criteria for comparing the performance of rule-based controllers and state-feedback controllers.

2 Fuzzy Sets and Rule-based Controllers

A difficulty facing many applications of AI in control is how to handle *imprecision* in the knowledge expressed by expert controllers. Fuzzy set theory - suggested by Zadeh - provides the facility to express the imprecise knowledge by using *linguistic variables* [14]. We have argued elsewhere about the importance of handling different types of uncertainty in AI systems (e.g., [3],[4]).

The basic idea in fuzzy control centers around the labeling process, in which the reading of a sensor is translated into a label as done by human expert controllers. For example, in the context of controlling a nuclear reactor [1], an observed reactor period (i.e., the rate of rise of the power) might be classified as *too short*, *short*, or *negative*. It is important to note that the transition between the labels are continuous rather than abrupt. This means that a positive reactor's period of 90 seconds might be termed *too short* to degree 0.2, *short* to degree 1.0, and "negative" to degree 0.0 [1]. A similar concept is used in our experiment: an angular position of say 5 degrees might be called *Positive* to a degree of .8 and *Zero* (i.e., a label used to describe very small angles) to degree of 0.2. This idea of *partial matching* plays an important role in fuzzy control and is related to the concept of a membership function used in fuzzy set theory where the boundary of a set is not sharp and the *degree of membership* specifies how strongly an element belongs to a set.

The knowledge base of a fuzzy controller consists of rules which are described using *linguistic variables*. Since one or more sensor readings might trigger several control rules at the same time, a *conflict resolution* strategy is needed. We could use a *Max-Min compositional rule of inference* for conflict resolution which works as follows: Assume that we have the following two rules:

Rule 1: IF X_1 is A_1 and X_2 is B_1 THEN Y is C_1
Rule 2: IF X_1 is A_2 and X_2 is B_2 THEN Y is C_2

Now, if we have x_1 and x_2 as the sensor readings for fuzzy variables X_1 and X_2, then their *truth values* are represented by $A_1(x_1)$ and $B_1(x_2)$ respectively for Rule 1. Similarly for Rule 2, we have $A_2(x_1)$ and $B_2(x_2)$ as the truth values of the preconditions. Then what we call the *strength* of Rule 1 could be calculated by:

$$S_1 = A_1(x_1) \ \wedge \ B_1(x_2), \tag{1}$$

where $\wedge$ is a conjunction operator which is most often defined to be the min function. Hence,

$$S1 = \min(A_1(x_1), B_1(x_2)). \tag{2}$$

Similarly for Rule 2:

$$S_2 = A_2(x_1) \wedge B_2(x_2) = \min(A_2(x_1), B_2(x_2)). \tag{3}$$

The effect of the strength of Rule 1 on its conclusion is calculated by:

$$C_1' = S_1 \wedge C_1 = \min(S_1, C_1), \tag{4}$$

and for Rule 2:

$$C_2' = S_2 \wedge C_2 = \min(S_2, C_2). \tag{5}$$

This means that Rule 1 is recommending a control action with C_1' as its membership function and Rule 2 is recommending a control action with C_2' as its membership function. The conflict-resolution process treats these as disjunctions, i.e.:

$$C = C_1' \vee C_2' \tag{6}$$

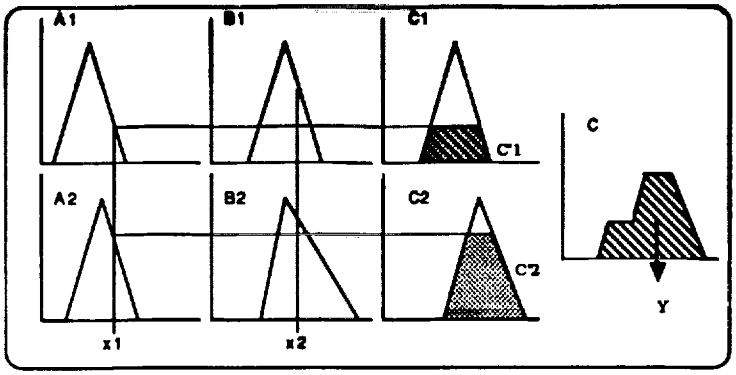

Figure 1: Conflict-resolution when two or more rules can fire

where $\vee$ is most often defined to be the max function. Hence,

$$C = \max(C_1', C_2'). \tag{7}$$

Since the result of this last operation itself is a membership function curve and a crisp control action is needed for the output, we can calculate the centroid of the area under the membership function. In mathematical terms this would mean:

$$Y = \frac{\int C(y) y dy}{\int C(y) dy} \tag{8}$$

This process is illustrated graphically in Figure 1.

3 Rule-based Control vs. Analytical Control

We offer the following general remarks when contrasting rule-based and analytical control. Section 5 provides a more detailed comparison:

- Analytical controllers and rule-based controllers are similar in the sense that both require that the control designer to have a detailed knowledge of the main parameters of the process. They differ in how they treat these parameters: analytical controllers need a precise mathematical model of the process, and the rule-based controllers need control rules from the experts but no precise mathematical model. Hence, one approach is more formal requiring deeper knowledge while the other is more heuristic in nature.

- Rule-based controllers are more robust to the variations in the initial conditions of the process and are more tolerant of sensor failures.

- Rule-based controllers require a long and tedious *calibration* process for fine-tuning the parameters of the control. In the case of fuzzy control, this usually means adjusting the applied membership functions.

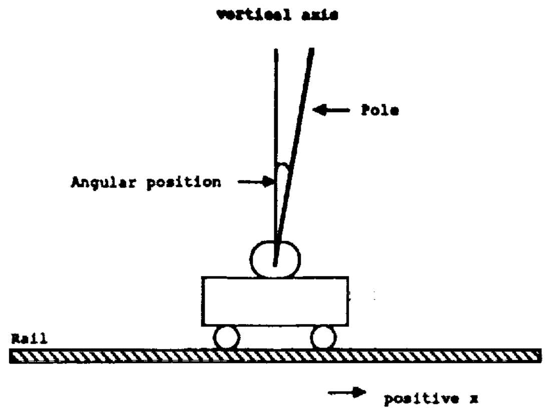

Figure 2: A schematic of the cart-pole balancing prototype hardware system

- Analytical controllers achieve the steady state faster than the rule-based controllers and are more sensitive to the signal noise.

We address the above points in our experiment with the cart-pole balancing problem described in the next section.

4 The Cart-Pole balancing problem

This section presents the POLE program, whose task is to balance a pole hinged to a motor-driven cart (see Figure 2). The cart moves on rail tracks to its right or its left depending on the instruction generated by POLE. The pole has only one degree of freedom (rotation about the hinge point). The primary control task of POLE is to balance the pole within a certain small range of cart positions on the rail and to keep it vertical.

Four state variables are used to describe the system status at each stage, and one variable represents the force applied to the cart. These are:

θ	:	angle of the pole with respect to the vertical line
$\dot{\theta}$	:	angular velocity of pole θ
x	:	horizontal position of the cart on the rail
$\dot{x}$	:	velocity of the cart
F	:	amount of force applied to the cart to move it toward the left or the right

Three labels are used to linguistically define the value of each of the state variables $\theta, \dot{\theta}, x$, and $\dot{x}$: Positive (PO), Zero (ZE), and Negative (NE). Figure 3(a) illustrates the memberships of these linguistic terms.

For force F, we use seven fuzzy labels: Negative-Small (NS), Negative-Medium(NM), Negative-Large (NL), Zero (ZE), Positive-Small (PS), Positive-Medium (PM), and Positive-Large (PL). Figure 3(b) illustrates the membership functions associated with these labels.

POLE is a rule-based program consisting of only 13 rules, nine of which are used to control the angular position and the others to control the position of the cart. The format

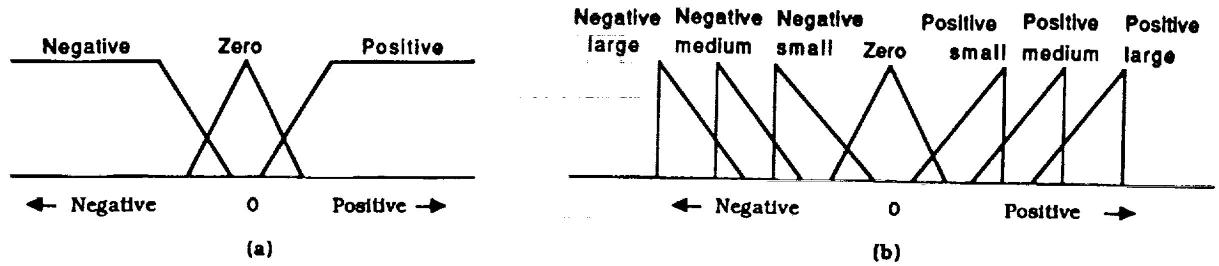

Figure 3: (a)- Three qualitative levels for $\theta, \dot{\theta}, x$, and $\dot{x}$, (b)- Seven qualitative levels for F

of the rules is simple, having two or four preconditions and one consequent. The main reason for the simplicity of these rules is that they are *fuzzy control rules*, and the terms in the preconditions can cover a large class of sensor readings, each to a different degree. A typical fuzzy control rule is:

IF θ is PO and $\dot{\theta}$ is ZE THEN F is PS

Appendix A gives a listing of the 13 control rules used by the program.

A characteristic of the POLE program, as well as some other systems based on fuzzy control, is that at any instant of time, more than one fuzzy rule might be ready to fire. Then POLE has to perform conflict-resolution using the heuristic *Max-Min rule of composition* explained in section 2.

5 Experiments

A set of 7 poles of different lengths and weights were used in our experiments. The length of these poles varied between 0.5 and 2 meters and their weights varied between 0.05 and 2.0 Kilograms:

Pole	Length (m)	Weight (Kg)	Pole	Length (m)	Weight (Kg)
1	1.0	0.100	5	1.0	0.500
2	0.5	0.050	6	1.0	1.000
3	1.0	0.050	7	1.0	2.000
4	0.5	0.025			

In each experiment, we compared the performance of the fuzzy controller with the performance of the state feedback controller. In each case, the fuzzy controller performed better with less under- and over-shoot. However, it took a longer period for the fuzzy controller to reach stability, especially for controlling the position of the cart. Table 1

	Pole-1		Pole-2		Pole-6	
	FC	SFC	FC	SFC	FC	SFC
Max. θ overshoot (degrees)	.33	1.00	.34	1.11	.25	.63
Max. θ undershoot (degrees)	.87	2.29	.73	2.41	.38	2.52
θ settling time (seconds)	3.5	4.2	3.00	4.8	5.3	3.00
Max. x overshoot (cm)	14.7	17.1	8.8	16.2	19.5	21.6
Max. x undershoot (cm)	.8	1.7	.5	2.9	1.6	0.0
x settling time (seconds)	38.2	4.6	41.9	6.5	45.9	2.8

Table 1: Comparison of the fuzzy controller (FC) and the State Feedback Controller (SFC)

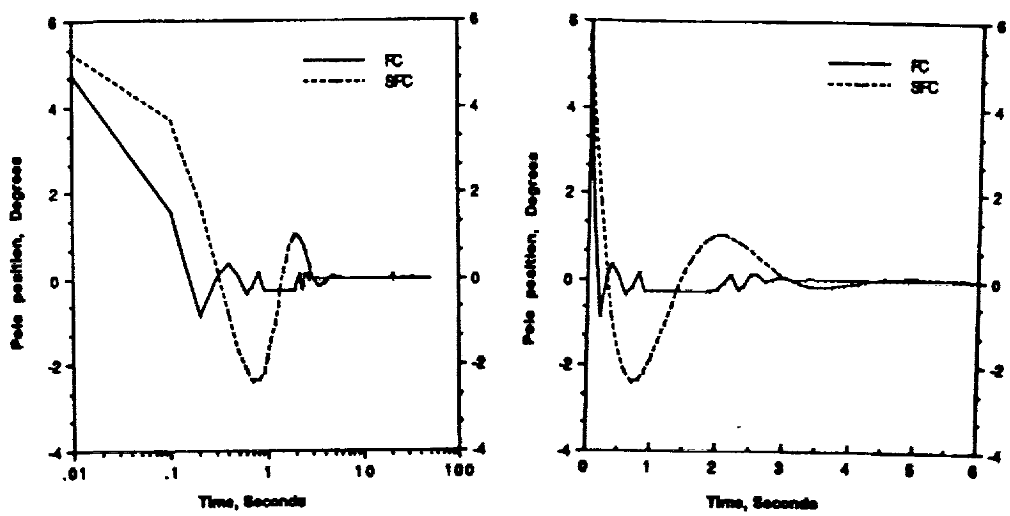

Figure 4: Angular control by Fuzzy Controller and State Feedback Controller

summarizes this difference for three of these poles, over a total sampling time of 50 seconds and a time step of 5 mili-seconds.

Figures 4, 5, present graphical display of the performance of the fuzzy controller (FC) and the state feedback controller (SFC) in controlling angular position (θ) and cart position (x) for Pole 1.

6 Criteria for performance comparison

The following criteria, derived from these experiments, can be used in deciding which type of controller to use in a given application.

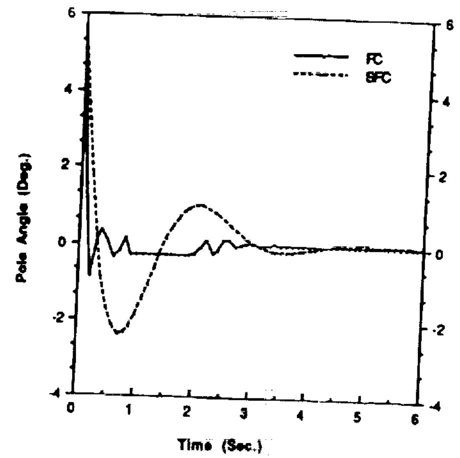

Figure 2: Simulation data - Pole Angular control

for the simplicity of these rules is that they are *linguistic control rules*, and the terms in the preconditions can cover a large class of sensor readings, each to a different degree.

A characteristic of the POLE program, as well as some other systems based on linguistic control, is that at any instant of time, more than one linguistic rule might be ready to fire. In this case, POLE performs conflict-resolution using the heuristic *Max-Min rule of composition* explained in Section 2.

5 Simulations and Experiments

In this section, the performance of POLE is compared with a State Feedback Controller (SFC). SFC is one of the modern control techniques [Kailath 80] which uses a control law $u = -kx$. u is the input variable of the physical system, which is a real number in single-input systems; x is the state variable which is an n-element column vector; k is the n-element row vector of feedback gains. The SFC formulation is based on the state space representation of the controlled system. The equations governing the cart-pole system are given in the Appendix B[‡].

Simulation-Based Comparison: We first tested the performance of these controllers using computer simulation. A set of 7 poles of different lengths and weights were used. The length of these poles varied between 0.5 and 2 meters and their weights varied between 0.05 and 2.0 Kilograms. We use the notation (Pole-#, Length(m), Weight(Kg)). The poles had the following characteristics: (Pole-1, 1.0, .1), (Pole-2, .5, .05), (Pole-3, 1.0, .05), (Pole-4, .5, .025), (Pole-5, 1.0, .5), (Pole-6, 1.0, 1.0), and (Pole-7, 1.0, 2.0).

In each experiment, we compared the performance of the fuzzy controller with the performance of the state feedback controller. In each case, the fuzzy controller performed better, with less under- and over-shoot. However, it took more time for the fuzzy controller to reach stability, especially for controlling the position of the cart. Table 1 summarizes this difference for three of the poles, over a total sampling time of 50 seconds and a simulation time step of 5 mili-seconds.

Figure 2 presents a graphical display of the performance of the fuzzy controller (FC) and the state feedback controller (SFC) in controlling the pole's angular position (θ).

[‡]Due to space limitations, we avoid describing the lengthy process of modeling and system identification which was required to design the SFC controller

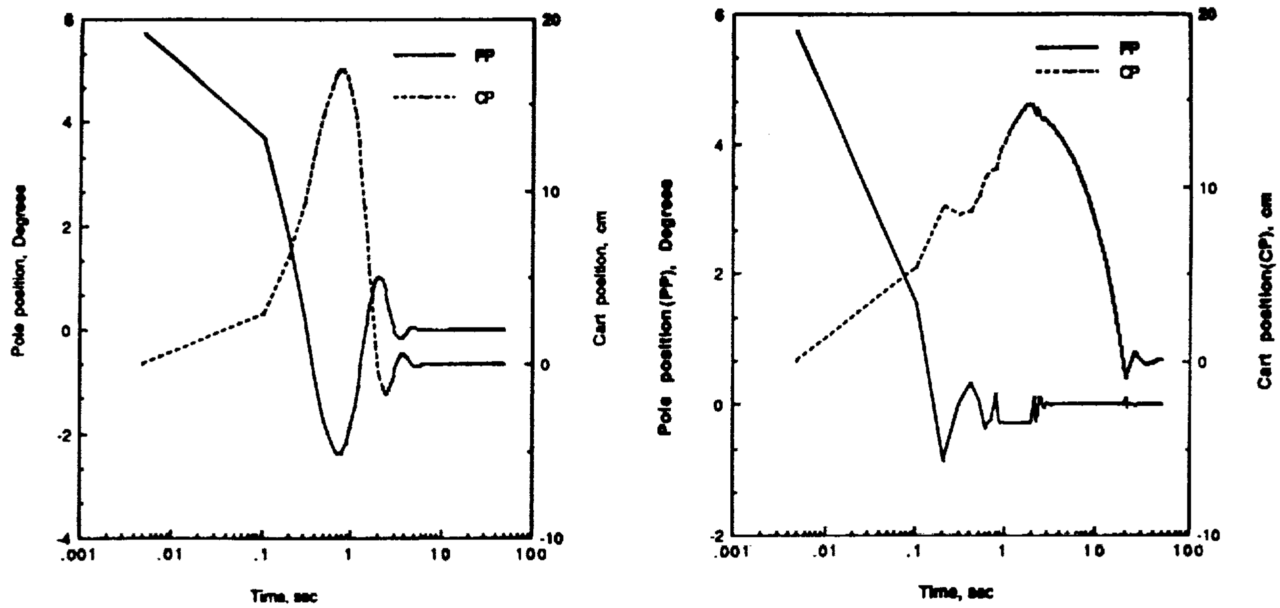

Figure 5: Angular and cart position control by Fuzzy and State Feedback Controllers

- Design complexity: This factor could well act as a decisive element in selecting an approach. For processes that are ill-defined and do not have physical mathematical models, rule-based fuzzy controllers should be considered. The *complexity* issue was not a problem in our experiment since analytical models were readily available for the cart-pole balancing problem. However, for a large class of non-linear control problems, this criterion has a significant role.

- Completeness: This is of concern mainly in design of rule-based controllers, but it applies as well to analytical models. The *completeness* requirement is that every possible state of the process must be accounted for by the knowledge base of the controller. Use of continuous membership functions for the predicates of the rules and multiple rules for each state met this criterion in our experiment.

- Robustness: The controller must be robust under a wide range of process parameters. In our experiments with poles of varying lengths and weights, the fuzzy controller was more robust than the analytical controller. In the case of Pole 7 (the heaviest and longest pole), the fuzzy controller balanced the pole, albeit with difficulty; the state feedback controller, however, failed to balance the pole at all.

- Performance: The relative importance of such performance parameters as steady-state accuracy, settling time, over-shoot and under-shoot depend on the application domain. In the experiments described in the last section, the fuzzy controller produced less under- and over-shoots than the state feedback controller. However, the state feedback controller attained the state state faster than the fuzzy controller.

- Modification of the controller: Depending on the domain, it might be desirable to update the knowledge used in the design of the controller (e.g., new technology development). Since rule-based fuzzy controllers need a significant calibration effort of adjusting the membership functions, modifying the knowledge base of the controller may require considerable effort.

7 Conclusions

The POLE program demonstrates an approach that uses approximate reasoning in control. We argue that this is an important factor in applying a human expert controller's knowledge. We used POLE and its prototype hardware development to compare the performance of a rule-based controller with that of an analytical state feedback controller. POLE produced results very close to its counterpart analytical controller and in some cases, POLE's results surpassed them. We believe that these results are good indications of the versatility of our approach in general as a *complement* to the conventional controllers. Also, in cases where a precise model of the process is not easily available, rule-based controllers are preferred.

More work has to be done on the POLE program to allow *automatic learning* of approximate control rules, not only by analyzing successes and failures (where the only inputs to the learning system are the binary failure/success modes) but also by analyzing the process trends. The inclusion of this capability would reduce the current long process of calibration needed to design rule-based controllers.

References

[1] Bernard, J. A., Use of a Rule-Based System for Process Control, *IEEE Control System Magazine*, October 1988.

[2] Berenji, H. R., Chen, Y. Y., Lee, C. C., Murugesan, S., The inverted- pendulum control using approximate reasoning– A short video presentation, *Uncertainty in AI workshop, American Association for Artificial Intelligence Conference*, St. Paul, 1988.

[3] Berenji, H. R., Responses to Saffioti's *treatment of Uncertainty in AI*, Knowledge Engineering Reviews, volume 3, no 2, 1988.

[4] Berenji, H. R., Treatment of Uncertainty in Artificial Intelligence, in: *Machine Intelligence for Aerospace Robotics*, Eds. E. Heer and H. Lum, AIAA, 1989.

[5] Chen, Y.Y., Stability Analysis of Fuzzy Control — A Lyapunov Approach, *proceedings of IEEE Systems, Man, and Cybernetics Annual Conference*, vol. 3, pp. 1027-1031, 1987.

[6] Connell, M. E., Learning to Control a Dynamic Physical System, AAAI, pp. 456-460, 1986.

[7] Lee, C. C., Fuzzy Logic in Control Systems: Fuzzy Logic Controller, University of California , Berkeley, 1988.

[8] Mamdani, E. H., and Assilian, S., An Experiment in linguistic synthesis with a fuzzy logic controller, *International Journal of Man-Machine Studies*, Volume 7, no.1, pp. 1-13, 1975.

[9] Michie D., and Chambers R., Boxes: An experiment in Adaptive Control, in *Machine Intelligence 2*, E. Dale and D. Michie Eds., Oliver and Boyd, Edinburg, 1968.

[10] Larkin, L.I., A fuzzy logic controller for aircraft flight control, *Industrial Applications of Fuzzy Control*, Sugeno (ed.), 1985.

[11] Ostergaard, J.J., Fuzzy logic control of a heat exchanger process, in Fuzzy Automata and Decision Processes, *North-Holand*, 1977.

[12] Selfridge, O. G., Sutton R.S., and Barto A. G., Training and Tracking in Robotics, in Proceedings of the 9th International Joint Conference on Artificial Intelligence, Cambridge MA, 1977.

[13] Shaefer, J. F., and Cannon, R. H., Continuous Liner Systems: On the Control of Unstable Mechanical Systems, 1966.

[14] Zadeh, L.A.,The concept of a linguistic variable and its application to approximate reasoning, *Inform. Sci.*, Part 1: vol. 8, pp. 199-249; Part 2: vol. 8, pp. 301-357; Part 3: vol. 9, pp. 43-80, 1975.

[15] Zadeh, L.A., Fuzzy Logic, *IEEE Computer*, pp. 83-93, 1988.

Appendix A: Controller's knowledge base

Two sets of fuzzy control rules have been used for the results in this paper. The first set contains 9 rules and is used for Angular position control. The second set contains 4 rules and is used for horizontal position control of the cart. The following notation is used in the presentation of these rules:

θ	:	Angular position	NE	:	Negative
$\dot{\theta}$	:	Angular speed	NL	:	Negative Large
x	:	Horizontal position of the cart	NM	:	Negative Medium
$\dot{x}$	:	Cart velocity	NS	:	Negative Small
PO	:	Positive	F	:	Force applied to the cart
PL	:	Positive Large	VS	:	Very Small
PM	:	Positive Medium	PVS	:	Positive Very Small
ZE	:	Zero	NVS	:	Negative Very Small

Rules used for angular position control:

RIA-88-06-02-1

Simulating Futures in Extended Common LISP
Philip Nachtsheim — June 1988

Stack-groups comprise the mechanism underlying implementation of multiprocessing in Extended Common LISP, i.e., running multiple quasi- simultaneous processes within a single LISP address space. On the other hand, the future construct of MULTILISP, an extension of the LISP dialect SCHEME, deals with parallel execution. The source of concurrency that future exploits is the overlap between computation of a value and use of the value. This paper describes a simulation of the future construct by an interpreter utilizing stack-group extensions to Common LISP.

RIA-88-08-01-6

Bayesian Classification
Peter Cheeseman, James Kelly, Matthew Self, John Stutz, William Taylor, and Don Freeman — August 1988

This paper describes a Bayesian technique for unsupervised classification of data and its computer implementation, AutoClass. Given real valued or discrete data, AutoClass determines the most probable number of classes present in the data, the most probable descriptions of those classes, and each object's probability of membership in each class. The program performs as well as or better than other automatic classification system when run on the same data and contains no *ad hoc* similarity measures or stopping criteria. Autoclass has been applied to several databases in which it has discovered classes representing previously unsuspecting phenomena.

RIA-88-11-01-1

Goal Ordering in Partially Ordered Plans
Mark Drummond and Ken Currie — November 1988

Partially ordered plans have not solved the goal ordering problem. Consider: a goal in a partially ordered plan is an operator precondition that is not yet achieved; operators, orderings and variable bindings are introduced to achieve such goals. While the planning community has known how to achieve *individual* goals for some time, there has been little work on the problem of *which* one of the many possible goals the planner would achieve next. This paper argues that partially ordered plans do not usefully address the goal-ordering problem and then presents a heuristic called *temporal coherence* which does. Temporal coherence is an admissible heuristic which provides goal-ordering guidance. Temporal coherence is admissible in the sense that if a solution exists in the planner's search space, then there will be a series of goal achievements permitted by the heuristic which can produce this solution.

RIA-88-12-05-1

Controlling a Dynamic Physical System With Approximate Reasoning
Hamid Berenji, Yung-Yaw Chen, Chuen-Chien Lee, and S. Murugesan — December 1988

This paper presents an approach for controlling a dynamic physical system by using approximate reasoning. The approach has been implemented in a program named POLE, and we have successfully built a prototype hardware system to solve the cart-pole balancing problem in real time. This provides a complementary alternative to the conventional analytical control methodology, and is of substantial use where a precise mathematical model of the process being controlled is not available. Also we furnish a set of criteria for comparing controllers based on approximate reasoning and those based on conventional control schemes.

REPORT DOCUMENTATION PAGE

OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
Dates attached

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
Titles/Authors - Attached

5. FUNDING NUMBERS

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Code FIA - Artificial Intelligence Research Branch
Information Sciences Division

8. PERFORMING ORGANIZATION REPORT NUMBER
Attached

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Nasa/Ames Research Center
Moffett Field, CA. 94035-1000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Available for Public Distribution
Peter Friedland 5/14/92 BRANCH CHIEF

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Abstracts ATTACHED

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT